Modul Hari Ketiga

Basic Android Application Development

Hanya untuk penggunaan pribadi dan proses pembelajaran di kelas Basic Android Application Development Codepolitan.com

1. Navigasi dengan ViewPager

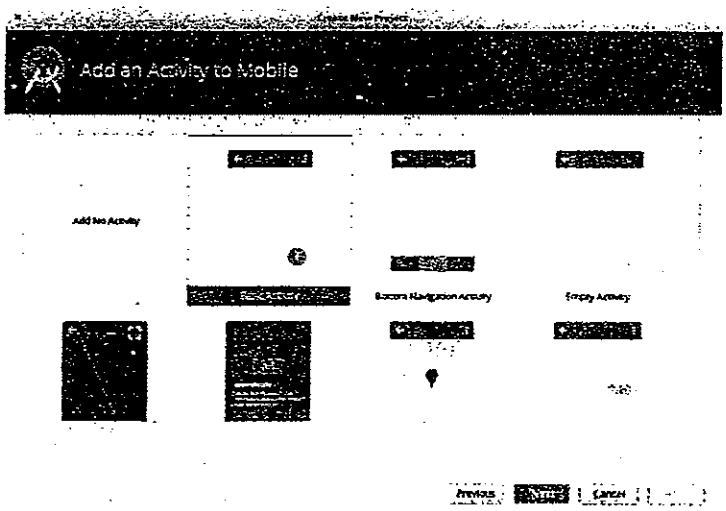
Kemarin kita sudah mempelajari bagaimana membuat sebuah navigasi sederhana dengan memanfaatkan Activity, Fragment, dan Button. Sekarang, mari kita belajar bagaimana membuat Navigasi yang sesuai dengan panduan Google dengan menggunakan ViewPager.

Buat project baru dengan nama **ViewPager**, minimum SDK level 16, namun dengan template **Basic Activity.**

Seperti ini tampilan aplikasi yang akan kita buat.

1.1. Struktur Basic Activity

Mari kita lihat struktur project yang baru dibuat.



Karena kita memilih template Basic Activity, disini kita memiliki tiga file yaitu:

- MainActivity.java

- activity_main.xml
- content_main.xml

Apabila memilih Basic Activity, kita akan selalu diberikan dua file xml seperti di atas. File activity_main.xml merupakan layout yang dibaca oleh MainActivity. Hal ini bisa kita pastikan dengan melihat method `setContent()` di onCreate:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // ...
}
```

Meski berbeda file, namun activity_main.xml dan content_main.xml masih dibaca sebagai satu keutuhan. Perhatikan baris perintah `<include ... />`, perintah ini akan membaca file content_main.xml dan memasukkannya ke dalam activity_main.xml. Struktur ini dibuat secara otomatis oleh Android Studio untuk merampingkan isi satu file. Kode-kode di dalam content_main.xml yang akan kita masukkan nanti bisa saja ditulis langsung di activity_main.xml.

1.2. Menambah Komponen ViewPager

Baiklah, sekarang kita lanjutkan dengan membuat sebuah komponen ViewPager di content_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:context="com.codepolitan.viewpager.MainActivity"
    tools:showIn="@layout/activity_main">

    <android.support.v4.view.ViewPager
        android:id="@+id/viewPager"
```

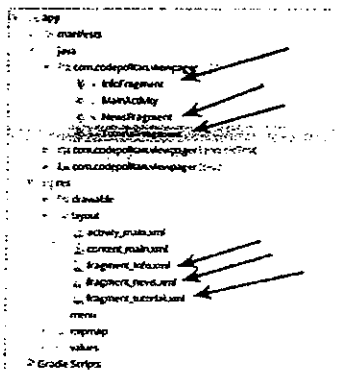
```
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</RelativeLayout>
```

Komponen `<ViewPager>` ini akan berfungsi seperti `<FrameLayout>` yang diberikan id container. Kelebihannya ialah, ia sudah memiliki kemampuan untuk mendeteksi *swipe* untuk melakukan perpindahan Fragment.

1.3. Buat Fragment

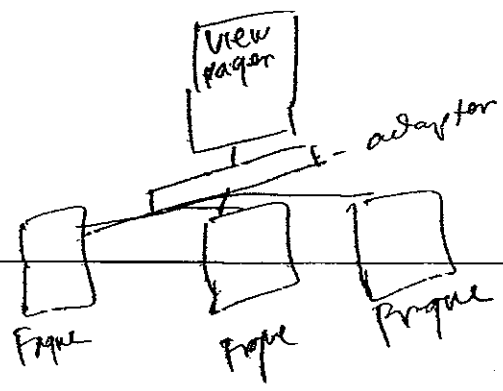
Setelah menambah komponen ViewPager, sekarang mari kita buat terlebih dahulu Fragment-Fragment isinya. Buat tiga buat Fragment dengan nama NewsFragment, TutorialFragment, InfoFragment. Ubah teks yang ada di TextView masing-masing layout untuk memudahkan kita memeriksa hasilnya.

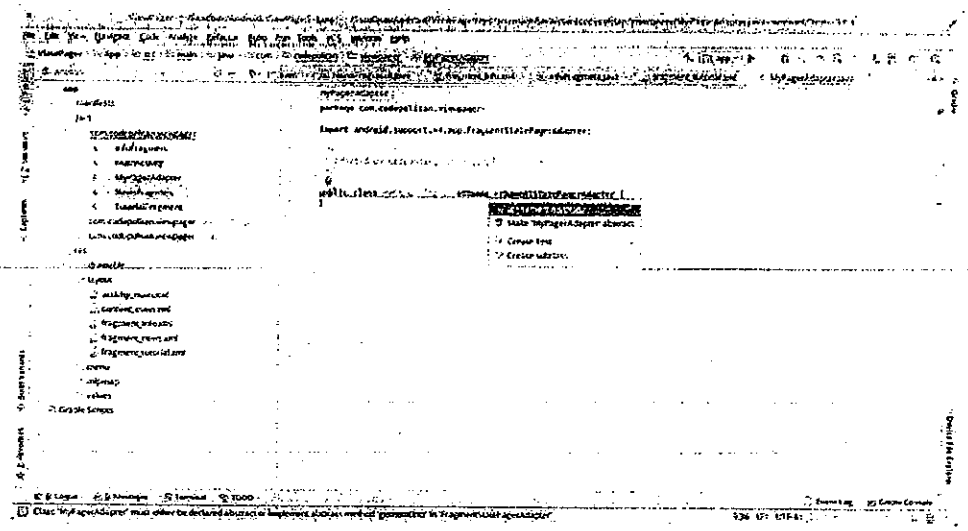


1.4. FragmentStatePagerAdapter

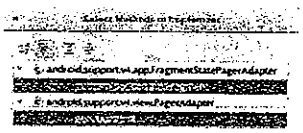
Untuk menampilkan Fragment-Fragment di sebuah ViewPager dan melakukan navigasi dengan swipe, kita perlu membuat sebuah kelas pembantu yang meng-extends FragmentStatePagerAdapter. Buat sebuah kelas kosong bernama MyPagerAdapter.java dengan cara klik kanan > New > Java Class.

Klik di baris yang berwarna merah, tekan ALT + ENTER lalu pilih **Implement methods** seperti pada gambar di bawah.



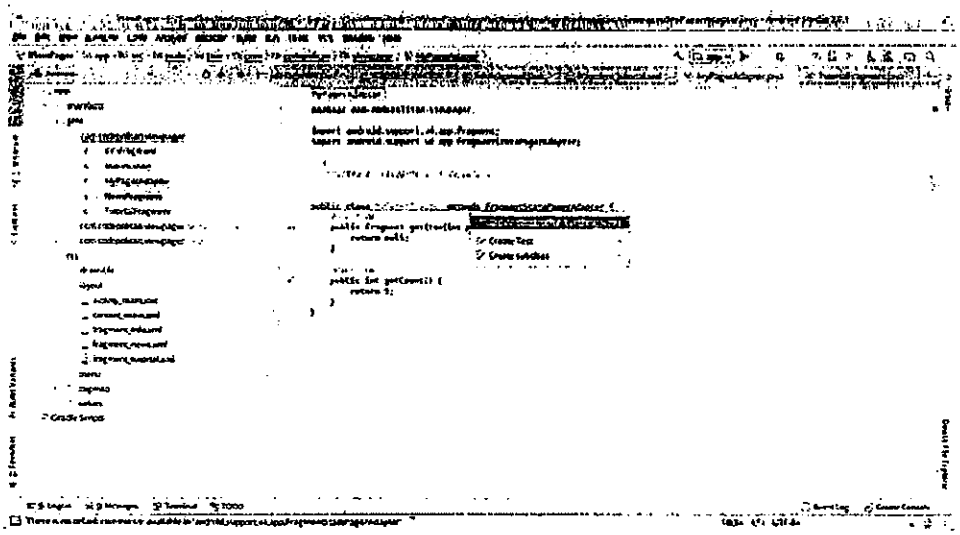


Pada jendela yang muncul pilih semua methods. Setelah tombol **OK** di klik, maka Android Studio akan membuatkan dua methods yang harus kita *implement* karena meng-extends kelas FragmentStatePagerAdapter





Selanjutnya kita perlu membuat sebuah constructor. Klik lagi di baris berwarna merah, tekan ALT + ENTER lalu pilih **Create constructor matching super**.



```
public class MyPagerAdapter extends FragmentStatePagerAdapter {

    public MyPagerAdapter(FragmentManager fm) {
        super(fm);
    }

    @Override
    public Fragment getItem(int position) {
        return null;
    }

    @Override
    public int getCount() {
        return 0;
    }

}
```

Di dalam method `getCount()` kita harus me-return berapa jumlah Fragment yang ingin ditampilkan.

```
@Override
public int getCount() {
    return 3;
}
```

Method `getItem()` akan dipanggil setiap kali user memilih salah satu Fragment dengan melakukan swipe (saat pertama kali dibuka secara otomatis berada di

posisi 0). Oleh karena ini di dalam method ini lah kita akan membuat objek Fragment sesuai urutan tampilannya. Misalnya disini kita ingin menampilkan NewsFragment diposisi pertama, diikuti oleh InfoFragment dan TutorialFragment.

1.5. Update MainActivity

Sekarang kita kembali ke MainActivity untuk menyelesaikan pembuatan ViewPager. Didalam method `onCreate()` kita perbarui isinya menjadi:

```
public class MainActivity extends AppCompatActivity {

    ViewPager myViewPager;
    MyPagerAdapter adapterViewPager;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        myViewPager = findViewById(R.id.viewPager);
        adapterViewPager = new
MyPagerAdapter(getSupportFragmentManager());

        myViewPager.setAdapter(adapterViewPager);
    }
}
```

Pertama yang kita lakukan adalah melakukan deklarasi objek ViewPager dan MyPagerAdapter. Lalu kita inisiasi kedua objek ini di `onCreate`.

Masih ingat dengan perintah `getSupportFragmentManager().beginTransaction().add().commit()`? Betul, alih-alih mengganti secara manual sebuah Fragment ke container, kita menyuruh sebuah adapter untuk melakukannya secara otomatis menggunakan komponen ViewPager.

Jalankan aplikasi ke device atau emulator lalu coba swipe ke kiri dan ke kanan. Bagiamana hasilnya?

1.6. Menambah TabLayout

Sampai di sini sebetulnya sistem navigasi sudah bisa digunakan. Namun, untuk melengkapi sistem navigasi menggunakan ViewPager, kebanyakan orang akan ikut menambahkan sebuah TabLayout. TabLayout digunakan untuk menampilkan judul dari masing-masing Fragment yang dapat ditampilkan oleh ViewPager.

TabLayout dapat diletakkan di mana, namun pada umumnya dan apabila mengikuti panduan dari Google, TabLayout akan diletakkan dibagian atas. Agar terlihat menyatu dengan `Toolbar`, kita akan menambahkannya di activity_main.xml di dalam `<AppBarLayout>`.

```
<android.support.design.widget.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:theme="@style/AppTheme.AppBarOverlay">

    <android.support.v7.widget.Toolbar
        android:id="@+id/toolbar"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="?attr/colorPrimary"
        app:popupTheme="@style/AppTheme.PopupOverlay" />

    <android.support.design.widget.TabLayout
        android:id="@+id/tabLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</android.support.design.widget.AppBarLayout>
```

Setelah memperbarui komponen XML, kini saatnya kita memperbarui MainActivity.java dan memberitahu TabLayout untuk terhubung dengan ViewPager.

```
public class MainActivity extends AppCompatActivity {

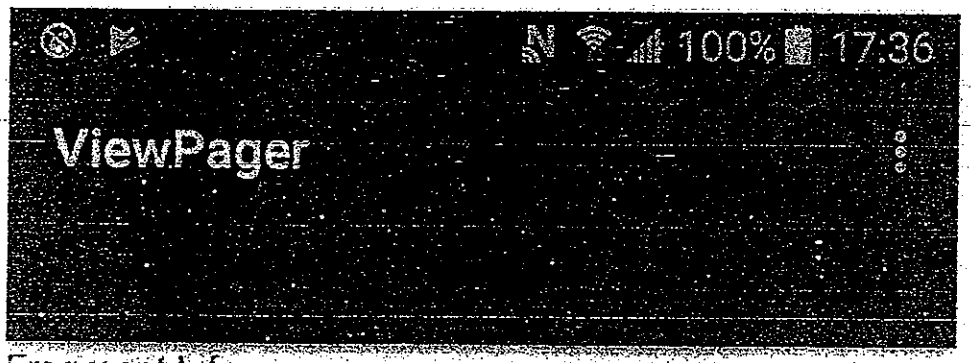
    ViewPager myViewPager;
    MyPagerAdapter adapterViewPager;
    TabLayout tabLayout;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        myViewPager = findViewById(R.id.viewPager);
        adapterViewPager = new
MyPagerAdapter(getSupportFragmentManager());

        myViewPager.setAdapter(adapterViewPager);

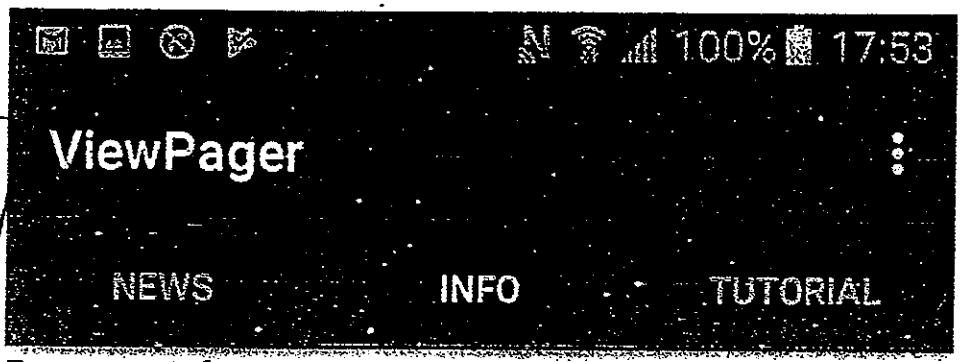
        tabLayout = findViewById(R.id.tabLayout);
        tabLayout.setupWithViewPager(myViewPager);
    }
}
```



Fragment Info

TabLayout sudah tampil, sudah dapat bekerjasama dengan ViewPager akan tetapi ia belum menampilkan judul untuk masing-masing Fragment sehingga dapat menyulitkan pengguna. Untuk menambahkan judul untuk masing-masing Fragment, tambahkan method berikut di MyPagerAdapter.java.

```
@Override
public CharSequence getPageTitle(int position) {
    if (position == 0) {
        return "News";
    } else if(position == 1) {
        return "Info";
    } else {
        return "Tutorial";
    }
}
```



Fragment Info

2. Menampilkan List dengan RecyclerView

Sebuah aplikasi Android belum lengkap rasanya jika belum menampilkan kumpulan data dalam bentuk List. Untuk menampilkan list di Android kita bisa memanfaatkan ListView, GridView, dan RecyclerView. Untuk data yang sederhana ListView/GridView memang lebih mudah dipelajari, tetapi ketika datanya lebih rumah implementasinya tidak jauh berbeda dengan RecyclerView sehingga di kelas ini kita akan langsung belajar cara menggunakan RecyclerView.

Kita akan melanjutkan project ViewPager sebelumnya dengan menampilkan sebuah list di masing-masing Fragment.

2.1. Menambahkan Dependensi

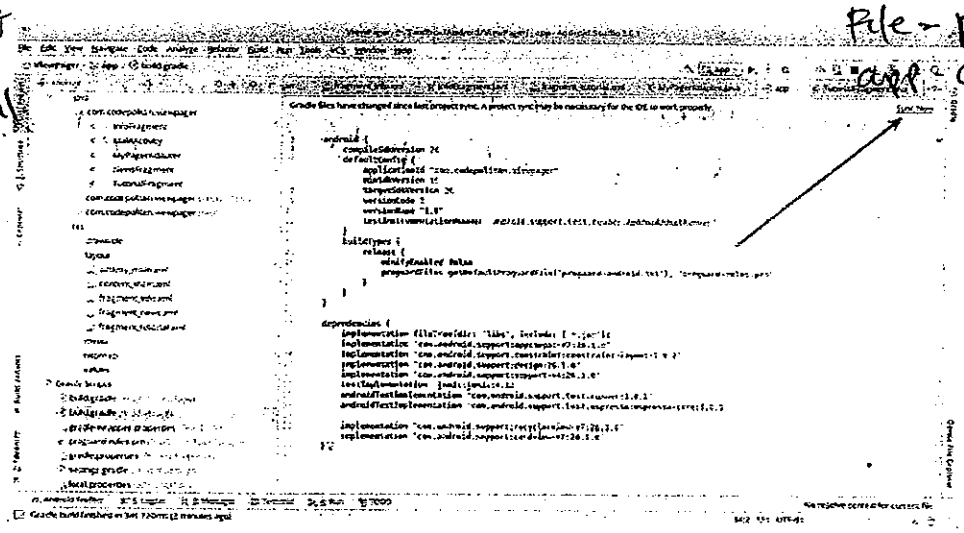
Pertama kita perlu menambahkan dependensi RecyclerView dan CardView di file **res > build.gradle**:

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'com.android.support:appcompat-v7:26.1.0'
    implementation 'com.android.support.constraint:constraint-
layout:1.0.2'
    implementation 'com.android.support:design:26.1.0'
    implementation 'com.android.support:support-v4:26.1.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'com.android.support.test:runner:1.0.1'

 androidTestImplementation 'com.android.support.test.espresso:espresso-
core:3.0.1'

    implementation 'com.android.support:recyclerview-v7:26.1.0'
    implementation 'com.android.support:cardview-v7:26.1.0'
}
```

Setiap kali mengubah file **build.gradle** kita perlu me-refresh-nya dengan mengklik tombol **Sync now**.



2.2. Menambah Layout XML

Setelah menambah dependensi RecyclerView dan CardView, mari kita tambahkan dulu kedua komponen ini. Buka fragment_news.xml lalu ubah menjadi:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
android"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.codepolitan.viewpager.NewsFragment">

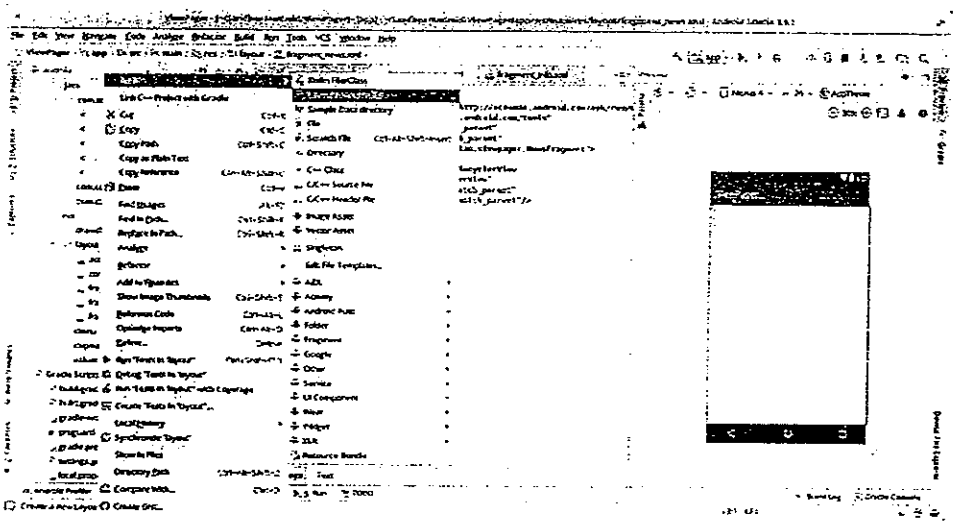
    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</RelativeLayout>
```

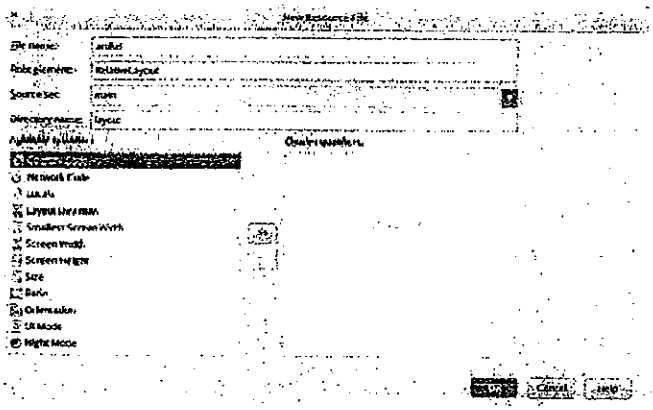


Perlu diingatkan lagi, saat ini kita menampilkan konten lewat Fragment bukan lewat Activity. Peran Activity saat ini hanya sebagai inang Fragment-Fragment yang kita buat (sebagai tempat navigasi ViewPager).

Kemudian kita buat sebuah layout baru untuk menampilkan item-item di dalam list. Klik kanan di folder **res > layout** lalu pilih **New > Layout resource file**.



Kita beri nama artikel dan ubah **root layout** menjadi RelativeLayout.



Isikan dengan kode berikut:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.v7.widget.CardView xmlns:android="http://
schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="16dp"
    >

    <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
android"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >

        <ImageView
            android:id="@+id/ivSampul"
            android:layout_width="match_parent"
            android:layout_height="200dp"
            android:scaleType="centerCrop"
            android:src="@drawable/ic_launcher_background" />

        <TextView
            android:paddingLeft="8dp"
            android:textStyle="bold"
            android:textSize="20sp"
            android:text="Ini adalah Judul"
            android:id="@+id/tvJudul"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/ivSampul" />

        <TextView
```

```
            android:paddingLeft="8dp"
            android:paddingBottom="8dp"
            android:text="Nama penulis"
            android:id="@+id/tvPenulis"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/tvJudul" />
    </RelativeLayout>

</android.support.v7.widget.CardView>
```

2.3. Membuat Kelas Data (POJO)

Sebelum membuat kumpulan data, kita harus membuat sebuah kelas POJO (Plain Old Java Object). Apa itu kelas POJO? Kelas POJO adalah sebuah kelas Java yang merepresentasikan data bentuk data kiat.

Buat sebuah kelas Java baru bernama Artikel.java dan tuliskan kode berikut:

```
public class Artikel {

    String id;
    String slug;
    String title;
    String author_name;
    String author_image;
    String description;
    String date;
    String link;
    String thumbnail;
    String total_views;

    public Artikel() {
    }

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getSlug() {
```

```
        return slug;
    }

    public void setSlug(String slug) {
        this.slug = slug;
    }

    public String getTitle() {
        return title;
    }

    public void setTitle(String title) {
        this.title = title;
    }

    public String getAuthor_name() {
        return author_name;
    }

    public void setAuthor_name(String author_name) {
        this.author_name = author_name;
    }

    public String getAuthor_image() {
        return author_image;
    }

    public void setAuthor_image(String author_image) {
        this.author_image = author_image;
    }

    public String getDescription() {
        return description;
    }

    public void setDescription(String description) {
        this.description = description;
    }

    public String getDate() {
        return date;
    }

    public void setDate(String date) {
        this.date = date;
```

```
    }

    public String getLink() {
        return link;
    }

    public void setLink(String link) {
        this.link = link;
    }

    public String getThumbnail() {
        return thumbnail;
    }

    public void setThumbnail(String thumbnail) {
        this.thumbnail = thumbnail;
    }

    public String getTotal_views() {
        return total_views;
    }

    public void setTotal_views(String total_views) {
        this.total_views = total_views;
    }

    public Artikel(String id, String slug, String title, String
author_name, String author_image, String description, String date,
String link, String thumbnail, String total_views) {
        this.id = id;

        this.slug = slug;
        this.title = title;
        this.author_name = author_name;
        this.author_image = author_image;
        this.description = description;
        this.date = date;
        this.link = link;
        this.thumbnail = thumbnail;
        this.total_views = total_views;
    }
}
```

2.4. Membuat Kumpulan Data

Setelah memiliki kelas data/POJO, sekarang saatnya kita membuat kumpulan data sederhana (sebelum kita isi dengan mengambil data dari sebuah API). Buka file FragmentNews.java dan tambahkan kode berikut:

```
public class NewsFragment extends Fragment {

    ArrayList<Artikel> kumpulanArtikel = new ArrayList<>();

    public NewsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {
        View v = inflater.inflate(R.layout.fragment_news, container,
false);

        for (int i = 0; i < 10; i++) {
            kumpulanArtikel.add(new Artikel(
                    "21769",
                    "python-software-foundation-terima-hibah-170000-
dari-mozilla-open-source-program-5a275ad53f531",
                    "Python Software Foundation Terima Hibah $170.000
dari Mozilla Open Source Program",
                    "Lord Voldemort",
                    "https://www.gravatar.com/
avatar/6a7d1f244946a619883181f5b6f0cddd?s=100",
                    "Beberapa waktu yang lalu, Python Software
Foundation (PSF) menerima hibah $170.000 dari Mozilla Open Source
Program. Hibah ini diberikan oleh..",
                    "06 Dec 2017",
                    "http://www.codepolitan.com/python-software-
foundation-terima-hibah-170000-dari-mozilla-open-source-
program-5a275ad53f531",
                    "https://static.cdn-
cdpl.com/270x135/998b78e349061b4971c0a2b0e8d6be41/header-python-
logo-flat-image700x350-crop-image700x350-crop-1-image(700x350-crop)-
image(700x350-crop).png",
```

```
                ""));
        }

        return v;
    }
}
```

2.5. Inisiasi RecyclerView

Setelah menyiapkan kumpulan data, selanjutnya kita lakukan inisiasi RecyclerView. Masih di FragmentNews.java tambahkan kode berikut:

```
public class NewsFragment extends Fragment {

    ArrayList<Artikel> kumpulanArtikel = new ArrayList<>();
    RecyclerView recyclerView;

    public NewsFragment() {
        // Required empty public constructor
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                             Bundle savedInstanceState) {

        ... kode lain disembunyikan agar ringkas

        recyclerView = v.findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
LinearLayoutManager(getContext()));

        return v;
    }
}
```

Inisiasi RecyclerView sama saja dengan komponen XML lainnya. Deklarasi dulu objeknya lalu lakukan `findViewById()`. Sebuah RecyclerView dapat menampilkan datanya dalam bentuk Linear, Grid, atau StaggeredGrid. Kita akan menampilkan datanya dalam bentuk Linear (menyusun ke bawah).

Sampai di sini data kita belum akan tampil di RecyclerView mengapa? Karena RecyclerView belum tahu data apa saja yang harus ia tampilkan. Untuk itu sebuah RecyclerView membutuhkan sebuah adapter (seperti saat kita membuat ViewPager tadi).



Constructor LinearLayoutManager meminta sebuah context. Jika berada di kelas Activity kita bisa menuliskan NamaActivity.this namun jika berada di sebuah Fragment, kita bisa tuliskan `getContext()` yang akan memberian hasil yang sama.

2.6. Membuat Adapter RecyclerView

Untuk membuat sebuah adapter yang kita perlukan, buat sebuah kelas Java baru dan beri nama ArtikelListAdapter lalu tulis kodenya menjadi seperti berikut ini:

```
public class ArtikelListAdapter extends
 RecyclerView.Adapter<ArtikelListAdapter.ArtikelViewHolder> {

}
```

Membuat ViewHolder

Di dalam kelas ArtikelListAdapter buat sebuah kelas baru bernama ArtikelViewHolder sesuai dengan yang kita tulis pada potongan kode di atas:

```
public class ArtikelListAdapter extends
 RecyclerView.Adapter<ArtikelListAdapter.ArtikelViewHolder> {

    public class ArtikelViewHolder extends RecyclerView.ViewHolder {
    }

}
```



Setiap kelas Adapter RecyclerView harus meng-extends RecyclerView.Adapter<> dan kelas ViewHolder harus meng-extends RecyclerView.ViewHolder.

Kelas ViewHolder tidak wajib berada di satu file, boleh juga ditulis di kelas/file terpisah.

Di dalam kelas ViewHolder ini kita kita deklarasi komponen XML yang ada di `artikel.xml` dan melakukan `findViewById()`:

```
public class ArtikelViewHolder extends RecyclerView.ViewHolder {
    TextView tvJudul, tvPenulis;
    ImageView ivThumbnail;

    public ArtikelViewHolder(View itemView) {
        super(itemView);
        tvJudul = itemView.findViewById(R.id.tvJudul);
        tvPenulis = itemView.findViewById(R.id.tvPenulis);
        ivThumbnail = itemView.findViewById(R.id.ivSampul);
    }
}
```

Buat Kumpulan Data dan Constructor Adapter

Masih di kelas ArtikelListAdapter, tambahkan kode berikut:

```
public class ArtikelListAdapter extends
 RecyclerView.Adapter<ArtikelListAdapter.ArtikelViewHolder> {

    ArrayList<Artikel> data;
    Context context;

    public ArtikelListAdapter(ArrayList<Artikel> data, Context context)
    {
        this.data = data;
        this.context = context;
    }

    public class ArtikelViewHolder extends RecyclerView.ViewHolder {
        // ... isi dihilangkan agar lebih ringkas
    }

}
```

Kumpulan data di kelas AdapterListAdapter akan menerima kumpulan data yang tadi kita buat di NewsFragment. Lalu objek context akan menerima nilai dari `getContext()` yang nanti akan kita pakai untuk beberapa keperluan.

Implement Method RecyclerView.Adapter

Selanjutnya kita wajib meng-implement onCreateViewHolder, onBindViewHolder, dan getItemCount seperti berikut ini:

```
public class ArtikelListAdapter extends
 RecyclerView.Adapter<ArtikelListAdapter.ArtikelViewHolder> {

    // ... kode sebelumnya dihilangkan agar lebih ringkas

    @Override
    public ArtikelViewHolder onCreateViewHolder(ViewGroup parent, int
viewType) {
        return null;
    }

    @Override
    public void onBindViewHolder(ArtikelViewHolder holder, int position)
{

    }

    @Override
    public int getItemCount() {
        return 0;
    }

    public class ArtikelViewHolder extends RecyclerView.ViewHolder {
        // ... isi dihilangkan agar lebih ringkas
    }
```

onCreateViewHolder

Selanjutnya kita akan membaca layout yang tadi kita buat (`artikel.xml`) agar bisa menampilkan isi dari `data`.

```
@Override
public ArtikelViewHolder onCreateViewHolder(ViewGroup parent, int
 viewType) {
    View view = LayoutInflater.from(context).inflate(R.layout.artikel,
 parent, false);
```

```
return new ArtikelViewHolder(view);
```

onBindViewHolder

Objek yang kita return dari onCreateViewHolder akan masuk ke dalam onBindViewHolder sehingga kita bisa memanfaatkan objek TextView dan ImageView yang kita buat di ArtikelViewHolder.

Karena kita akan menampilkan gambar dari sebuah internet, kita membutuhkan sebuah *library* bernama Picasso. Tambahkan dependensi Picasso berikut ke file **build.gradle**:

```
compile 'com.squareup.picasso:picasso:2.5.2'
```

Lalu tulis method `onBindViewHolder` menjadi seperti berikut ini:

```
@Override
public void onBindViewHolder(ArtikelViewHolder holder, int position) {
    Artikel artikel = data.get(position);

    holder.tvPenulis.setText(artikel.getAuthor_name());
    holder.tvJudul.setText(artikel.getTitle());

    Picasso.with(context).load(artikel.getThumbnail()).into(holder.ivThumbnail);
}
```

getItemCount

Adapter yang kita buat harus tahu ada berapa jumlah data yang dimiliki. Kita bisa memanfaatkan method `size()` milik sebuah `ArrayList` untuk mendapatkan ukuran yang dinamis.

```
@Override
public int getItemCount() {
    return data.size();
}
```

Pembuatan Adapter telah selesai. Sekarang mari kita hubungkan objek RecyclerView di NewsFragment dengan Adapter ini.

2.7. Show Me The Money

Perbarui isi NewsFragment menjadi sebagai berikut:

```
public class NewsFragment extends Fragment {

    ArrayList<Artikel> kumpulanArtikel = new ArrayList<>();
    RecyclerView recyclerView;
    ArtikelListAdapter adapter;

    public NewsFragment() {
        // Required empty public constructor
    }


    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
 container,
                             Bundle savedInstanceState) {

        View v = inflater.inflate(R.layout.fragment_news, container,
 false);

        for (int i = 0; i < 10; i++) {
            kumpulanArtikel.add(new Artikel(
                    "21769",
                    "python-software-foundation-terima-hibah-170000-
dari-mozilla-open-source-program-5a275ad53f531",
                    "Python Software Foundation Terima Hibah $170.000
 dari Mozilla Open Source Program",
                    "Lord Voldemort",
                    "https://www.gravatar.com/
avatar/6a7d1f244946a619883181f5b6f0cddd?s=100",
                    "Beberapa waktu yang lalu, Python Software
 Foundation (PSF) menerima hibah $170.000 dari Mozilla Open Source
 Program. Hibah ini diberikan oleh..",
                    "06 Dec 2017",
                    "http://www.codepolitan.com/python-software-
foundation-terima-hibah-170000-dari-mozilla-open-source-
program-5a275ad53f531",
```

```
                "https://static.cdn-
cdpl.com/270x135/998b78e349061b4971c0a2b0e8d6be41/header-python-
logo-flat-image700x350-crop-image700x350-crop-1-image(700x350-crop)-
image(700x350-crop).png",
                ""));
        }

        recyclerView = v.findViewById(R.id.recyclerView);
        recyclerView.setLayoutManager(new
 LinearLayoutManager(getContext()));

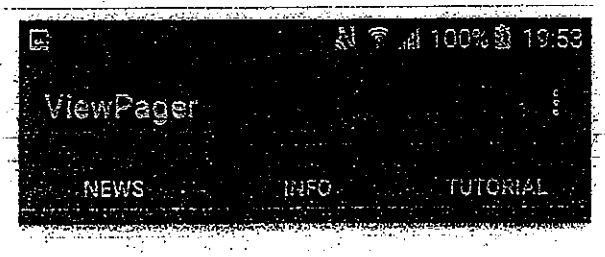
        adapter = new ArtikelListAdapter(kumpulanArtikel, getContext());
        recyclerView.setAdapter(adapter);

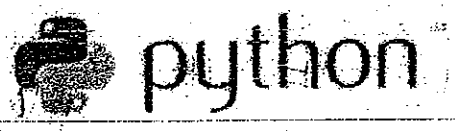
        return v;
    }

}
```

Yes, aplikasi kita sudah berjalan dengan baik dan menampilkan datanya kecuali gambar yang masih belum keluar. Karena kita menggunakan *library* Picasso yang akan mengunduh gambar dari sebuah URL, maka kita perlu menambahkan *permission* INTERNET. Buka file AndroidManifest.xml lalu tambahkan baris berikut di atas application:

```
<uses-permission android:name="android.permission.INTERNET"/>
```





Python Software Foundation Terima Hibah $170.000 dari Mozilla Open Source Program

Lord Voldemort

